# Multiple Realization and the Computational Mind

[Published in Proceedings of the AISB'11 Symposium on Computing and Philosophy, pp. 37-42, ISBN 978-1-908187-03-1]

# Multiple Realization and the Computational Mind

**Paul Schweizer**[1]

**Abstract.** The paper examines some central issues concerning the Computational Theory of Mind (CTM) and the notion of instantiating a computational formalism in the physical world. I address a standard line of criticism of CTM, based on the claim that the notion of instantiating a computational formalism is overly liberal to the point of vacuity, and conclude that Searle's view that computation is not an intrinsic property of physical systems is ultimately correct. I argue that for interesting and powerful cases, realization is only ever a matter of approximation and degree, and interpreting a physical device as performing a computation is relative to our purposes and potential epistemic gains. However, while this may fatally undermine a computational explanation of conscious experience, I contend that, contra Putnam and Searle, it does not rule out the possibility of a scientifically justified account of propositional attitude states in computational terms.

## 1 FORMALISM AND ARTEFACT

From an abstract mathematical perspective, computation comprises an extremely well defined and stable phenomenon. Central to the theory of computation is the intuitive notion of an effective or 'mechanical' procedure, which is simply a finite set of instructions for syntactic manipulations that can be followed by a machine, or by a human being who is capable of carrying out only very elementary operations on symbols. A key constraint is that the machine or the human can follow the rules without knowing what the symbols *mean*. The notion of an effective procedure is obviously quite general – it doesn't specify what form the instructions should take, what the manipulated symbols should look like, nor precisely what manipulations are involved. The underlying restriction is simply that they are finitary and can proceed 'mindlessly' i.e. without any additional interpretation or understanding. So there are any number of different possible frameworks for filling in the details and making the notion rigorous and precise. Turing's 'automatic computing machines' [1] (TMs), supply a very intuitive and elegant rendition of the notion of an effective procedure. But there is a variety of alternative frameworks, including Church's Lambda Calculus, Gödel's Recursive Function Theory, Lambek's Infinite Abacus Machines, etc.

According to the widely accepted Church-Turing thesis, the class of computable functions is captured in an absolute sense by the notion of TM computability, and compelling 'inductive evidence' for the thesis is supplied by the fact that every alternative formalization so far given of the broad intuitive notion of an effective procedure has been demonstrated to be equivalently powerful, and hence to specify exactly the same class of functions [2]. Thus the idealized notion of in-principle computability, where all finite bounds on input size, storage capacity and length of running time are abstracted away, seems to constitute a fundamental category, a stable and highly pleasing 'mathematical kind'.

A related further question to ask is whether any sort of comparable feature carries over to computation as implemented or realized in the physical universe. Turing machines and other types of computational formalisms are *mathematical abstractions*. Like equations, sets,

[1] Institute for Language, Cognition and Computation, School of Informatics, Univ. of Edinburgh, EH8 9AD, UK. Email: paul@inf.ed.ac.uk.

Euclid's perfectly straight lines, etc., TMs don't exist in real time or space, and they have no causal powers. In order to perform *actual* computations, an abstract Turing machine, thought of as a formal program of instructions, must be realized or instantiated by a suitable arrangement of matter and energy. And as Turing observed long ago [3], there is no privileged or unique way to do this. Like other abstract structures, such as chess games and isosceles triangles, Turing machines are *multiply realizable* - what unites different types of physical implementation of the same abstract TM is nothing that they have in common as physical systems, but rather a structural isomorphism in terms of a particular level of description. Hence it's possible to implement the very same computational formalism using modern electronic circuitry, a human being executing the instructions by hand with paper and pencil, a Victorian system of gears and levers, as well as more atypical arrangements of matter and energy including toilet paper and beer cans. Let us call this 'downward' multiple realizability, wherein, for any given formal procedure, this *same* abstract computational formalism can be implemented via an arbitrary number of *distinct* physical systems. And let us denote this type of downward multiple realizability as '↓MR'.

After the essential foundations of the mathematical theory of computation were laid, the vital issue then became one of engineering – how best to utilize state of the art technology to construct rapid and powerful physical implementations of the abstract mathematical blueprints, and hence perform actual high speed computations *automatically*. This is a clear and deliberate ↓MR endeavour, involving the intentional construction of artefacts, painstakingly designed to follow the algorithms that we have created. From this top-down perspective, there is an obvious and pragmatically indispensible sense in which the hardware that we have designed and built can be said to perform genuine computations in physical space-time.

## 2 COMPUTATION IN NATURE

In addition to these comparatively recent engineering achievements, but presumably still members of a single underlying category of phenomenon, various authors and disciplines propound the notion of 'natural computation' (NC), and invoke a host of indigenous processes as cases in point, including neural computation, DNA computing, biological evolution, molecular and membrane computing, slime mould growth, ant swarm optimization, 'embedded and pervasive computation', etc. According to such views, computation in the physical world is not merely artificial – it is not restricted to the devices specifically designed and constructed by human beings. Instead, computation is a seemingly ubiquitous feature of the natural order, and the artefacts invented by us constitute only a very small subset of the class of computational systems in the physical world.

The disciplinary and terminological practices surrounding NC invite a more thorough and rigorous examination of the underlying assumptions involved. To what extent is computation a genuine *natural* kind – is there any intrinsic unity or core of traits systematically held in common by the myriad of purported examples of computation in nature? This question has deep and independent conceptual significance, in an attempt to gain clarity on whether and to what extent computation can be cogently and fruitfully seen as a natural occurrence. In what sense, if any, can computation be said to take place spontaneously, as a truly native, 'bottom-up' phenomenon? And of course, the issue has special philosophical interest with respect to positions on the conjectured computational nature of *mentality and cognition*. It is this particular domain that will comprise the primary

focal point of the paper, within the broader context just outlined.

## 3 THE COMPUTATIONAL THEORY OF MIND (CTM)

According to the widely embraced 'computational paradigm', which underpins cognitive science, Strong AI and various allied positions in the philosophy of mind, computation (of one sort or another) is held to provide the scientific key to explaining and artificially reproducing mentality. The paradigm maintains that cognitive processes are essentially computational processes, and hence that intelligence in the physical world arises when a material system implements the appropriate kind of computational formalism. In terms of the classical model of computation as rule governed symbol manipulation, the relation between the abstract program level and its realization in physical hardware then yields an elegant solution to the traditional mind-body problem in philosophy: the *mind* is to the *brain* as a *program* is to the *hardware* of a digital computer.

It's an immediate corollary of CTM that the human brain counts as an exemplary instance of NC. However, CTM seems to require a more robust and literal stand on computation than that embraced by NC in general. It is crucial to recognize the distinction (as pointed out by, e.g. Gualtiero Piccinini [4]) between being a system/process that can be effectively *simulated* or *modelled* using a computational formalism and being a system/process that *literally instantiates* a computational procedure or executes an algorithm. Most purported cases of 'natural computation' in a scientific context are versions of the former and not the latter. It is clear that the brain *can* be viewed as a case of NC in this simulational or modelling sense. However, I take it that serious proponents of CTM would advocate a more substantive position, *viz*., that human mentality arises because the brain literally instantiates computational procedures and transforms symbol structures in a manner comparable to a computational artefact rather than a computer simulated thunderstorm.

According to CTM, mental states and properties are seen as complex internal processing states, which computationally interact within a system of internal state transitions, thereby mediating the inputs and outputs of intelligent behaviour. Hence any mental process leading to an action will have to be embodied as a physical brain process that realizes the underlying computational formalism. A perceived virtue of this approach is that it can potentially provide a *universal* theory of cognition, a theory which is not limited by the details and peculiarities of the human organism. Since mentality is explained in computational terms, and, as above, computational formalisms are multiply realizable, it follows that the mind-program analogy can be applied to any number of different types of creatures and agents. Combining CTM with ↓MR, it follows that a human, a Martian and a robot could all be in exactly the *same* mental state, where this sameness is captured in terms of implementing the same cognitive computation, albeit via radically different forms of physical hardware. So on this view, computation is seen as providing the scientific paradigm for explaining mentality in general – all cognition is to be literally described and understood in computational terms.

## 4 ANYTHING COMPUTES EVERYTHING

But rather than viewing ↓MR as a theoretical virtue promising a universal account of mentality, opponents of CTM target ↓MR as its Achilles heel. In *Representation and Reality*, Hilary Putnam [5] argues that implementing a computational formalism cannot serve as the theoretical criterion of mentality, because such a standard is overly liberal to the point of vacuity. As a case in point he offers a proof of the thesis

that *every* open physical system can be interpreted as the realization of *every* finite state automaton. In a related vein, John Searle [6] argues that computation is not an intrinsic property of physical systems. Instead, it an observer relative interpretation that we project on to various physical systems according to our interests and goals.

Searle contends that this makes CTM vacuous, because virtually any physical system can be interpreted as following virtually any program. Thus hurricanes, our digestive system, the motion of the planets, even an apparently inert lecture stand, all possess a level of description at which they instantiate any number of different programs – but it is absurd to attribute mental states and intelligence to them on that basis. Even though the stomach has inputs, internal processing states and outputs, it isn't a cognitive system. Yet if one wanted to, one could interpret the inputs and outputs as code for any number of symbolic processes. And in his article 'Is the Brain a Digital Computer' [7] Searle attempts to illustrate the extreme conceptual looseness of the notion of implementing an abstract formalism by famously claiming that the molecules in his wall could be interpreted as running the word star program.

Let us label multiple realizability in this direction, wherein any given *physical system* can be interpreted as implementing an arbitrary number of different *computational formalisms* 'upward MR' and denote it as '↑MR'. The basic import of ↑MR is the *non-uniqueness* of computational ascriptions to particular physical systems. In the extreme versions suggested by Putnam and Searle, there are apparently no significant constraints whatever – it is possible in principle to interpret every open physical system as realizing every computational procedure. Let us call this extreme version 'universal upward MR' and denote it as '↑MR*'. If every physical system can be construed as implementing every computational formalism, then clearly every computational formalism is realized by every physical system, and the corresponding position in the other direction, i.e. ↓MR*, is also true. So in this sense the two positions are equivalent and ↑MR* = ↓MR*.

But mere ↑MR is weaker than ↑MR*, since the former does not assert that there are no salient constraints, and hence ↑MR would be consistent with the denial that, e.g., the molecules in Searle's wall can in fact be interpreted as implementing the word star program, if we place the proper qualifications on the notion of implementation (although every physical system might still be interpretable as implementing some very large set of distinct computations). What ↑MR denies is simply that any particular computational description that can be legitimately applied is somehow privileged or unique.

## 5 SOME CONTRAINTS ARE IN ORDER

In response to the Putnam/Searle universal realizability objection, various defenders of CTM attempt to deny ↑MR* by (i) placing greater constraints on what counts as a legitimate physical realization and (ii) narrowing the set of computations relevant, since only very complex and advanced procedures will be of interest to CTM as candidates for mental architecture. Putnam's proof involves *inputless* finite state automata, and these are commonly dismissed as too primitive. Full input/output capabilities are required, as well as rich internal processing structure, which calls for something on a par with, say, Jerry Fodor's [8] Language of Thought (LOT) model of cognition.

In line with strategy (i) above, David Chalmers [9] advocates what he takes to be two essential constraints in distinguishing many of the 'false' cases of implementation assumed by Putnam's argument, from 'true' cases consistent with a non-trivial reading of CTM. The first is an appropriate *causal* structure relating the state transitions in the physical implementation of the

computational formalism (this is also proposed by, e.g. Ronald Chrisley [10]) , and the second is the ability of the mapping to support *counterfactual* sequences of transitions on inputs not actually given (which is also considered by Tim Maudlin [11]). Both of these are quite significant features inviting extended analysis, which unfortunately is not possible within the confines of the current discussion. However, selected points regarding each of these proffered constraints will be touched on below.

Chalmers argues that it is a necessary condition that the pattern of abstract state transitions constituting a particular run of the abstract computation on a particular input must map to an appropriate transition of physical states of the machine, where the relation between succeeding states in this sequence is governed by proper causal regularities. However, I would argue that this constraint is too strong in the general case. For example, in the Chinese room scenario, or indeed *any* situation where a human being is following an abstract computational procedure, the transition from one state to the next is not causal in any straightforward physical or mechanical sense. When I take a machine table set of instructions specifying a particular TM and then perform a given computation with pencil and paper by sketching the configuration of the tape at each step in the computation, the transitions sketched on the piece of paper are *not* causally connected: one sketch in the sequence in no way causes the next. It is only through my understanding and intentional choice to execute the procedure that the next state appears on the paper. Physical causation comes in only very indirectly, as in light rays illuminating the page and allowing me to see the symbols, and at an elementary and extraneous level, as in the friction between the pencil lead and the paper's surface causing various marks to appear.

Yet this is a perfectly legitimate and indeed paradigmatic case of implementing a Turing machine. In the Chinese room, it is merely through Searle's *understanding* of English, his free choice to behave in a certain manner, and a number of highly disjointed physical processes (finding bits of paper in a certain location, turning the pages in the instruction manual, all mediated by the human agent) that the implementation takes place. In this case it counts as an implementation simply because what can be interpreted as the appropriate states in the procedure *occur* in the correct linear order. Questions regarding the mechanics of *how* they happen to occur are not relevant to answering the question of whether or not the procedure has been implemented. The physical how is a *different* question, and is not on the same level of analysis as that invoked when determining whether or not the desired mapping from formalism to physical configuration obtains. But this then critically loosens the requirements for counting a physical system as instantiating a program. As long as what can be described or interpreted as the correct sequence of states actually occurs, then the underlying mechanics of how this takes place are not strictly relevant.

The causal requirements advocated by Chalmers constitute a sufficient but not a necessary condition – in the general case we must still allow for chance and human agency to play a role. However, the right sort of causal regularities and connections are needed if the instantiation in question is to be *fully automatic*, and if we want to be able to rely on the automatic device to perform systematically correct computations yielding outputs with the potential to supply us with new information. And although this is the norm when constructing and interpreting computational artefacts, it does not exhaust the general space of possibilities.

In response to Chalmers' proposed *counterfactual* requirement, it is worth noting that for a physical system to realize a rich computational formalism with proper input and output capacities, such as an abstract TM, this will always be a matter of *approximation*. For example, any given physical device will have a finite upper bound on the size of input strings it

is able to process, its storage capacities will likewise be severely limited, and so will its actual running time. In principle there are computations that formal TMs can perform which, even given the fastest and most powerful physical devices we could imagine, would take longer than the lifespan of our galaxy to execute. It will never be possible to construct a complete physical realization of an abstract TM – the extent to which the device can execute the full range of state transitions of which the abstraction is capable will always be a matter of *degree*. So in turn, the class of counterfactual cases on alternative inputs with which the realization can cope is by necessity limited – not all counterfactual cases will be supported by *any* physical device implementing a TM.

Consequently, there is no simple or principled cut off point demarking 'genuine' implementations from 'false' ones in terms of counterfactual considerations. Take a standard pocket calculator that can intake numbers up to, say, 6 digits in decimal notion. Is this a 'false' realization of the corresponding algorithm for addition, since it can't calculate $10^6 + 10^6$? It's an approximate instantiation which is nonetheless exceedingly useful for everyday sums. It will always be a matter of degree how many counterfactuals can be supported, where a single run on one inputV is the degenerate case. Where in principle can the line be drawn after that? It's a matter of our purposes and goals as interpreters and epistemic agents, and is not an objective question about the 'true' nature of the physical device as an implementation. In some cases we might only be interested in the answer for a single input, a single run

Hence for a physical device to successfully 'perform a computation' is distinct from 'fully instantiating a computational formalism'**.** Performing a computation is an occurrent event, an actual sequence of physical state transitions yielding an output value, whereas instantiating a complete computational formalism is much more stringent and hypothetical, requiring appeal to counterfactuals, and as above, this will only obtain as a matter of degree. In light of this distinction, it is clearly possible for a physical device to successfully perform a computation *without* instantiating a complete computational formalism.

## 6 OBSERVER RELATIVITY

One of Searle's basic claims is the allied tenet that computation is not an 'intrinsic' property of physical systems – instead it's an observer relative act of interpretation. This basic point has been objected to in different ways, and is itself in need of clarification. The latter part of Searle's claim may seem to suggest that it is a purely subjective matter, and Ned Block [12] objects by pointing out that it's simply not the case that anything goes. As an illustration, he notes that, although it's possible to reinterpret an inclusive OR gate as an AND gate by flipping our interpretations of the values of '0' and '1', it's simply not possible to reinterpret an *inclusive* OR gate as an *exclusive* OR gate. So although we have a great deal of latitude about how we interpret a device, there are also very important restrictions on this freedom, and according to Block, this makes it a substantive claim that, e.g., the human brain is a computer of a certain sort.

Block's position suggests that there are two important strands here that need to be separated. 'Observer relative' could mean that it's totally subjective and anything goes, which is the claim he wants to deny. But it could also mean something more curtailed, *viz*., that the attribution of computational activity requires an observer to supply the interpretation. This doesn't mean that the interpretation doesn't have to satisfy various objective constraints supplied by the given characterization of the system. It simply means that, as Searle also says, it's *not intrinsic* to the system itself, and must be provided by the observer as an outside ascription. Hence it's easy to reinterpret an inclusive OR gate as an AND gate – there is no objective fact to the matter as to which truth

function is being computed, and this is in perfect accord with ↑MR. Some interpretations appear to be excluded (on the very pivotal assumption that the physical system itself is characterized as an 'inclusive OR gate' and not as something more fundamental), which seems to cast some doubt on ↑MR*. In the present discussion I will not argue for or against ↑MR* (see Mark Bishop [13], [14] for an interesting version of the claim) but instead confine my considerations to the more modest ↑MR.

In view of ↑MR, it's still never the case that any given computational interpretation of a physical system is privileged or unique, and this seems far more difficult to deny than ↑MR*. And the non-intrinsic nature of computation is a direct consequence of ↑MR. As long as there are at least two distinct interpretations, there is no objective fact of the matter regarding *which* computation is being performed, and it follows that the computation itself is not an intrinsic property of the physical device. Instead, it is an act of human interpretation, and is usually tethered to issues involving design and engineering, relative to our purposes and interests. Thus implementation is always a matter of both interpretation and degree of approximation, and its usefulness will depend on our interests and epistemic needs (e.g. as above - how big a counterfactual set of inputs we want it to be able to compute).

It's certainly true that there is no pragmatic value in most interpretive exercises compatible with ↑MR and ↑MR*, e.g. *post hoc* attributions of single runs, or any case where we know the outcomes in advance of the interpretation. Physically instantiated computation is *useful* to us only insofar as it supplies informative outputs, which in most cases will come down to new information acquired as a result of the implemented calculation. Interesting observer relative computation takes place when we can directly read-off something that *follows from* the formalism, but which we didn't already know in advance and explicitly incorporate into the mapping from the start. That's the incredible value of our computational artefacts, and it's the only *practical* motivation for playing the interpretation game in the first place

Of course, this doesn't mean that we cannot ascribe other interpretations to the same system – the difference is that in most cases the outputs will then be of no pragmatic or epistemic value to us. But this is still something relative to our human interests, practices and goals – the success of the strategy is based on objective features of the system (typically that we have designed and built), but this does not make computation itself intrinsic – it is still an interpretation, an *abstract* level of description, and as such is neither canonical nor unique. Indeed, computation is no more an intrinsic property of a physical systems than is 'being a sequence of inscriptions constituting a formal derivation of a theorem in first-order logic'.

In line with this logic/formal proof example, when I execute a particular TM computation by drawing the initial tape configuration on a piece of paper, then write down the tape configuration for each step in the computation according to the instructions in the machine table until I reach a halting configuration and stop, the physical states realizing the computation are a sequence of scratch marks on a two dimensional sheet of paper. There is nothing *physical* about these scratched in patterns that is intrinsically computational – indeed, the shapes could be interpreted in any manner one likes or not at all. The computational interpretation of the physical scratch mark is purely *extrinsic*. And this is the same for syntactic interpretations in general – e.g. being an instance of the spoken English sentence 'The cat is on the mat' is not an intrinsic property of the sound waves constituting an instantiating utterance.

Physical systems as such are intrinsically rule (i.e. physical law) *obeying* while formal systems are intrinsically rule *following*. In the case of our computational artefacts, a rule obeying system must be deliberately engineered

so that it can be interpreted as isomorphic in the relevant sense to a chosen rule following formal system. Rule obeying is an essentially *descriptive* matter and there is no sense in which mistakes or error can be involved – physical law cannot be broken, and the time evolution of natural systems is wholly determined (in the classical case at least) by the laws obeyed. Rule following on the other hand is an essentially *normativ*e matter and there is a vital sense in which error and malfunction can occur. If my desk top machine is dosed with petrol and set on fire while still in operation, the time evolution of the hardware will remain in perfect descriptive accord with natural law. However, it will very soon fail to comply with the normative requirements of implementing Microsoft Word, and serious computational malfunctions will ensue. Being an implementation of Microsoft Word is a normative and *provisional* interpretation of the hardware system, which can be withdrawn when something goes 'wrong' or when the system is disrupted by non-design intended forces - being an implementation of Microsoft Word is not intrinsic to the physical structure itself.

## 7 COMPUTATION AND CONSCIOUSNESS

Many versions of CTM focus solely on the functional analysis of propositional attitude states such as belief and desire, and simply ignore other aspects of the mind, most notably consciousness and qualitative experience – Fodor's LOT is a classic case in point. However others, such as William Lycan [15], try to extend the reach of Strong AI and the computational paradigm, and contend that *conscious states* arise via the implementation of the appropriate computational formalism. This then invites reapplication of the Putnam/Searle line in the ↓MR* direction, with the rejoinder that every open physical system implements the 'appropriate computational formalism', so that consciousness is everywhere. According to this polemical strategy, rampant panpsychism follows as a consequence of CTM extended to the explanation of consciousness (which will be dubbed 'CTM+'), and this is taken as a *reductio ad absurdum* refutation of such views.

A natural line of defense for CTM+ is to invoke the counterfactual constraint above in order to deny ↓MR*. Only highly sophisticated physical systems (such as brains, presumably) are able to support all the counterfactuals required to count as an implementation of the appropriate computational formalism, and hence the attempted *reductio* is blocked. But as Maudlin and Bishop have argued, this is a highly dubious strategy in the case of conscious states, sense these are essentially *occurrent* phenomena, and the invocation of non-occurrent process seems to verge on the occult. As Bishop rightly observes, the appeal to counterfactuals seems to require a non-physical link between non-entered states and the resulting conscious experiences of the system.

And I would agree that for conscious states counterfactuals don't matter – it's only the *actual* run that could have any bearing, so that the foregoing attempted defense of CTM+ is unsuccessful. Additionally, I would argue that the computational account of consciousness is fundamentally wrong in any case, and that even given the implementation of all purportedly relevant counterfactuals, this would still not constitute a sufficient condition for the presence of conscious experience. As above, computation is not an intrinsic property of physical systems, and so is inherently unsuited to serve as the foundation for conscious experience, which should be based on intrinsic properties of the brain as a physical system. As I've argued elsewhere ([16], [17]), propositional attitudes are potentially explainable in terms of functional/computational structure, which is abstract and multiply realizable (because non-intrinsic!). In contrast, conscious states, if they occur in a given implementation, should be explained in terms of the intrinsic physical properties of the medium of instantiation.

This is because, unlike computational formalisms, conscious states are inherently *non-abstract*; they are *actual*, occurrent phenomena extended in physical time. The computational camp makes a critical error by espousing ↓MR as a hallmark of their theory, while at the same time contending that qualitatively identical conscious states are maintained across wildly different kinds of physical realization. The latter is the claim that an actual, substantive and *invariant* phenomenon is preserved overly radically diverse real systems, while the former is the claim that *no* internal physical regularities need to be preserved. And this implies that there is no actual, internal property that serves as the causal substrate or supervenience base for the substantive, invariant phenomenon in question. The advocate of CTM+ cannot rejoin that it is *formal role* which supplies this basis, since formal role is abstract, and such abstract features can only be *instantiated* via actual properties, but they do not have the power to *produce* them. The only (possible) non-abstract effects that instantiated formalisms are required to preserve must be specified in terms of their input/output profiles, and thus *internal* experiences, qua actual events, are in principle omitted. Hence it would appear that the actual, occurrent nature of conscious states entails that they must depend upon intrinsic properties of the *physical* world.

## 8 OBSERVER RELATIVITY AND CTM

However, content laden propositional attitudes *are* highly dispositional in character, and for such abstract, dispositional states, the relevant counterfactuals pertaining to formal processing structure *do* matter. If we restrict CTM to the belief-desire framework commonly assumed to characterize intentional systems, and leave consciousness out of its purview, then it is possible to give an account of how this type of approach could, at least in principle, offer us an effective theoretical handle on the mind. If we take something like Fodor's LOT (as a starting point for the sake of illustration), this is at least the basic type of highly sophisticated and complex computational structure relevant to CTM. Propositional attitudes themselves are abstract, dispositional states, and their functional/computational rendition could in principle be interpreted as a computational level of description of the activities of the human brain.

In line with the foregoing discussion, even if, for the sake of argument, we grant that the brain implemented Fodor's LOT, still, this would *not* be an intrinsic property of the brain as a biochemical mechanism. Instead, it would be a scientifically fruitful and explanatorily powerful level of description, which could supply a unifying perspective that ties together actual brain function, seen as neurologically implementing relevant tokens of 'mentalese' symbols, and systematically manipulating these tokens in a manner consistent with the proffered computational formalism of LOT. This abstract level of description would then have to mesh with the salient input and output capabilities that we want to explain via this attribution of internal cognitive structure. So from a purely physical perspective, the inputs and outputs are various forms of energy bombarding the organism's surface and emanating from it, and are not intrinsically computational either. But on the non-intrinsic cognitive level, these would be viewed as instances of written and spoken language, for example. And when interpreted as such, this non-intrinsic syntactic level will correspond to the internal processing activity triggered by the incoming energy pulse, interpreted as, say, a sentence in an English conversation.

There would be no scientific interest in a mere *a hoc* mapping from LOT onto the brain (though in principle this may be possible, *a la* ↓MR*). Instead, there would be a myriad of pre-existing and empirically intransigent 'wet-ware' constraints that the mapping would have to satisfy, in order to correspond to the salient causal structure of brain activity as discovered by neuroscience. And as above, this would have

to conform with observed input and output patterns interpreted symbolically, to yield successful *predictions* of both new outputs given novel inputs, and predictions correctly describing new brain configurations entailed by the theory as realizations of the appropriate formal transformations required to produce the predicted output. This would be real science, with two primary levels of empirical constraint satisfaction and experimental testing and confirmation, to establish or refute the accuracy of the proposed theoretical mapping. Additionally, the linguistic interpretation of input and output signals would have to mesh with corresponding objects and states of affairs in the agent's environment, since in the human LOT case, we are studying and explaining an environmentally embedded system, and not a solipsistic syntax manipulator.

If this CTM project were to turn out successful, then the LOT would be as powerful and well confirmed as a scientific venture could hope to be, and the objection that computation is still not an 'intrinsic' property of the brain would fade into irrelevance. It is in virtue of all of these factors considered together that human cognition could be accounted for in computational terms, and not simply in virtue of the brain being (in-principle) interpretable as realizing the LOT, by appeal to a mapping that ignores these crucial factors.

## 9 CONCLUSION

In accord with Searle, computation should be viewed as an extrinsic, observer relative feature of physical systems. As such, it does not constitute a stable or independent natural kind. Various natural phenomena can be modelled or simulated using computational techniques, but this is to be distinguished from the notion that the system *itself* spontaneously instantiates and executes a formal procedure. Natural systems are essentially rule obeying, and computational modelling simulates this in a fundamentally descriptive manner. In contrast, formal procedures are essentially normative, rule following structures, and in principle this interpretation can be projected onto natural systems in an almost limitless variety of ways. However, *interesting and illuminating* cases of computation realized in the physical world will come down to a question of engineering, either artificial or perhaps biological (to attain a robust, informative, non-post-hoc, multiple constraint satisfying *degree* of fit as a level of description for a physical system).

It is conceivable that the human brain has been biologically engineered such that there exist interesting and informative levels of computational description in the above sense. Hence I would conclude that Searle's basic point against CTM is not well taken. Although CTM+ and a computational theory of consciousness are ruled out, in the case of propositional attitude states, the non-intrinsic status of computation does not trivialize predictively successful ascriptions of formal structure, and multiple realizability on its own does not render CTM empirically vacuous.

## REFERENCES


[1] A. Turing, 'On Computable Numbers, with an Application to the Entscheidungsproblem, *Proceeding of the London Mathematical Society*, (series 2), 42, 230-265, (1936).

[2] Boolos, G., Burgess, J.P. and Jeffrey, R.C., *Computability and Logic*, 5th edition, Cambridge University Press, (2007).

[3] Turing, A., 'Computing Machinery and Intelligence', *Mind* 59: 433- 460 (1950).

[4] Piccinini, C., 'Computational Modelling vs. Computational Explanation', *The Australasian Journal of Philosophy*, 85(1), 93-115, (2007).

[5] Putnam, H., *Representation and Reality*, MIT Press, (1988).

[6] Searle, J., 'Minds, Brains and Programs', *Behavioral and Brain Sciences* 3: 417-424, (1980).

[7] Searle, J., 'Is the Brain a Digital Computer?', *Proceedings of the American Philosophical Association*, 64, 21-37, (1990).

[8] Fodor, J., *The Language of Thought*, Harvester Press, (1975).

[9] Chalmers, D. J., 'Does a Rock Implement Every Finite-State Automaton?', *Synthese*, 108, 309-333, (1996).

[10] Chrisley, R. L., 'Why Everything Doesn't Realize Every Computation', *Minds and Machines*, 4, 403-420, (1994).
[11] Maudlin, T., 'Computation and Consciousness', *Journal of Philosophy*, 86, 407-432, (1989).
[12] Block, N., 'Searle's Arguments against Cognitive Science'. In J. Preston and J. M. Bishop *Views into the Chinese Room*, Oxford University Press, (2002).
[13] Bishop, J. M., 'Dancing with Pixies'. In J. Preston and J. M. Bishop *Views into the Chinese Room*, Oxford University Press, (2002).
[14] Bishop, J. M., 'Why Computers Can't Feel Pain', *Minds and Machines*, 19, 507-516, (2009).
[15] Lycan, W. G., *Consciousness*, MIT Press, (1987).
[16] Schweizer, P., 'Physicalism, Functionalism and Conscious Thought.' *Minds and Machines*, 6**,** 61-87 (1996).
[17] Schweizer, P., 'Consciousness and Computation.' *Minds and Machines*, 12, 143-144, (2002)